

Eight Years of Farming; Is Everybo(ar)dy Happy?

Embedded Linux Conference Europe 2024

Geert Uytterhoeven
geert@linux-m68k.org

<http://glider.be/>

September 17, 2024

© Copyright 2024 GLIDER bv, CC BY-SA 4.0 © ⓘ ⓘ



Table of Contents

Introduction

Part I: My Board Farm

Part II: Remote Access



2 / 39

About Me (and Linux)

- ▶ **Freelance Embedded Linux Kernel Hacker @ Glider BV**
- ▶ Started with Linux as a hobbyist, 30 years ago
- ▶ Amiga, m68k, PPC, MIPS, PS3/Cell, Renesas, ARM, Super-H, RISC-V, ...
- ▶ Networking, graphics, IDE, audio, RTC, clock, pin control, GPIO, ...
- ▶ **Maintainer of the m68k architecture**
since 2004
- ▶ **Maintainer of Renesas clock and pin control drivers**
since 2016
- ▶ **Maintainer of Renesas ARM SoC platforms**
since July 2019



3 / 39

Eight Years of Farming; Is Everybo(ar)dy Happy?

PART I: My Board Farm

ELCE2016 in Berlin: **Herd Your Boards, Become a Farmer***

- ▶ What has changed?
- ▶ What worked well?
- ▶ What didn't work well?

PART II: Remote Access

Open Up Your Board Farm, Securily

*https://elinux.org/ELC_Europe_2016_Presentations



4 / 39

My Board Farm

What is a Board Farm?

- ▶ Collection of SBCs and development boards in a central place
- ▶ Accessed remotely in a uniform way

What do I use it for? Why?

- ▶ (Remote) Development, Testing, Debugging
- ▶ Convenient access to lots of boards
- ▶ Keep my desk clean
- ▶ ☹ No physical access

What do other people use it for?

- ▶ Development, Testing, Debugging
- ▶ Automated testing
- ▶ No local board available
- ▶ Special requests



5/39

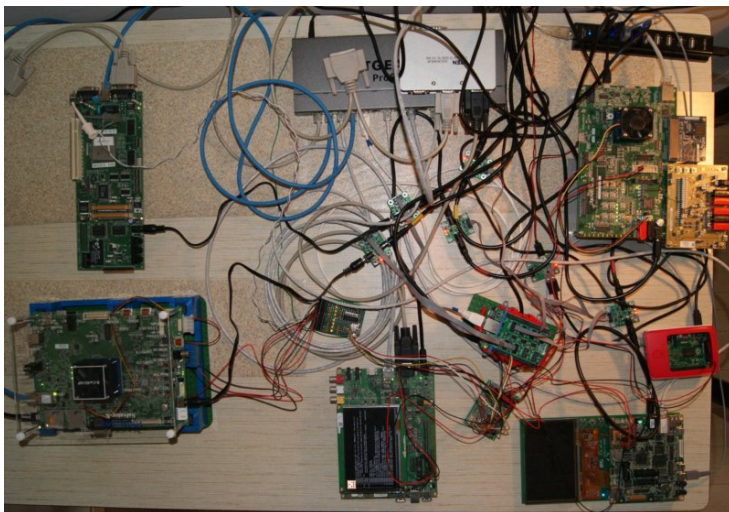
Back to 2016

- ▶ 6 boards
- ▶ PC Power Supply with Terminal Blocks and Fuse Holders
- ▶ BeagleBone Black (Standby Power)
- ▶ BayLibre ACME for Power Control/Measurement
- ▶ Custom Opto-Board for Reset/Keypresses/...
- ▶ 16-port Ethernet Switch
- ▶ 10-port USB 2.0 hub
- ▶ Quad USB-serial adapter



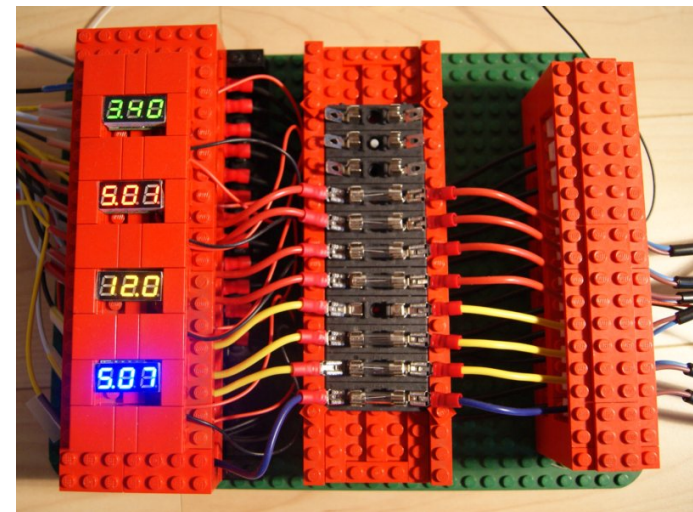
6/39

2016: Old Kitchen Table



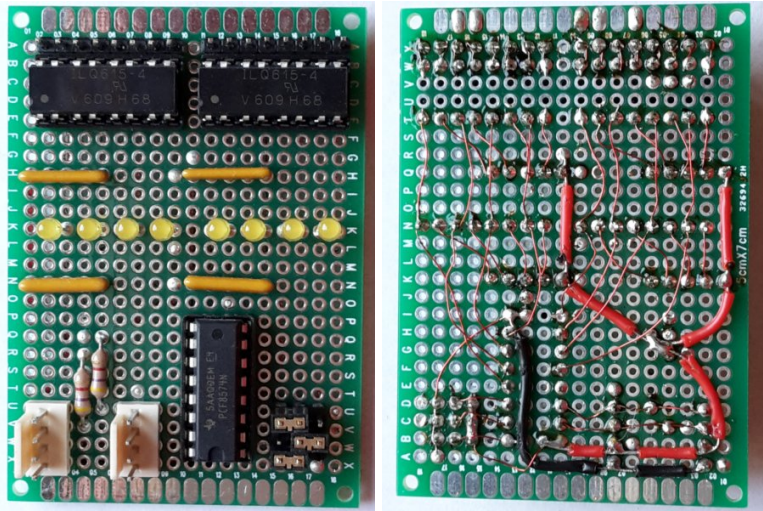
7/39

2016: Powered by LEGO

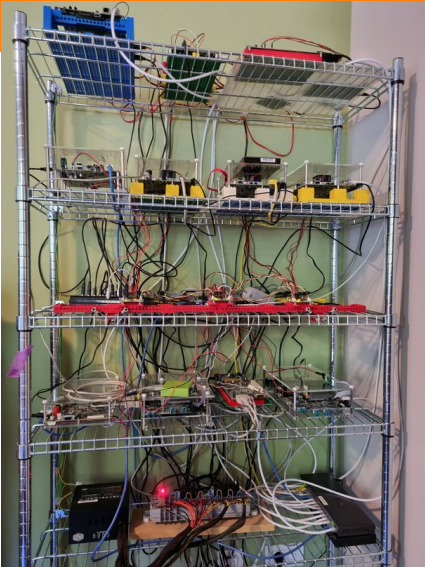


8/39

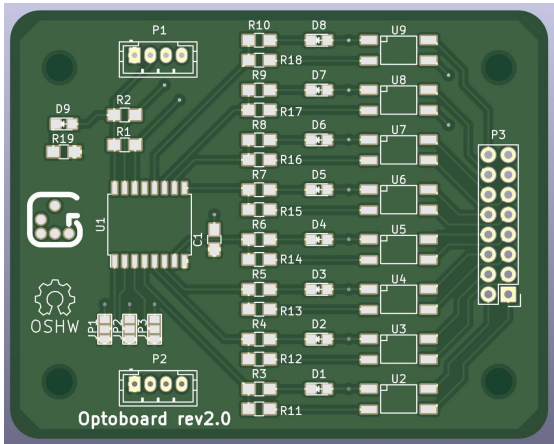
2016: Electronics Prototyping



Fast-Forward to 2024



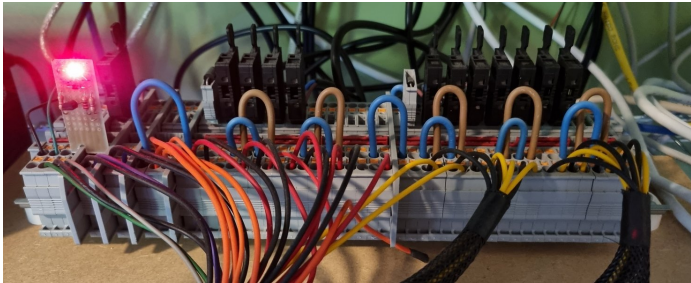
Let's Learn KiCad!



<https://github.com/geertu/Optoboard>

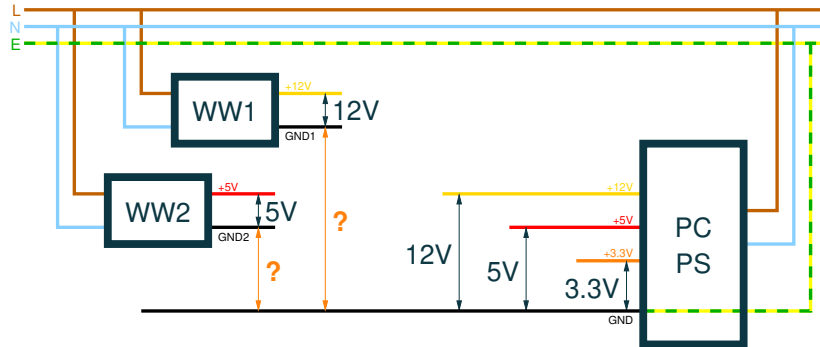
General Improvements

- ▶ Industry-Standard Terminal Blocks (Phoenix Contact) mounted on DIN rail
 - ▶ Easier to wire and extend
 - ▶ Fuse and component (TVS diode) holders



- ▶ Grounded Metal Shelves
- ▶ Smoke Detector

Avoiding Ground Loops



- ▶ Earth Ground \equiv Chassis Ground \equiv Signal Ground
- ▶ I2C works well for me



13 / 39

Adding More Boards (Take One)

Requirements / Design

Build my own All-in-One Board?

- ▶ Power Control/Measurement
- ▶ Opto-Isolators
- ▶ USB-Serial Adapter
- ▶ Some Intelligence

PJRC Teensy 3.2

- ▶ NXP MK20DX256
- ▶ 72 MHz ARM Cortex-M4
- ▶ 64 KiB RAM, 256 KiB FLASH
- ▶ Micro-USB
- ▶ 3 serial ports (2 with FIFO)
- ▶ I2C, GPIO, PWM

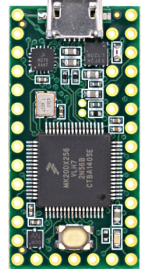
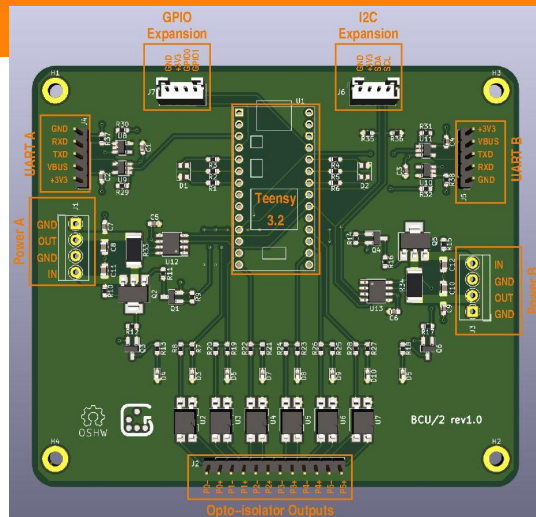


Image © PJRC



14 / 39

BCU/2: PCB



<https://github.com/geertu/pcb-bcu2>



15 / 39

BCU/2 — Board Control Unit Duo: Software

Challenge

Teensyduino supports one virtual serial port over USB (CDC-ACM)

Solution

Added support for Dual and Triple Serial (integrated upstream)

- ▶ Bare Metal, Event Loop, Cooperative Scheduler
- ▶ USB Host Connectivity, 3 Virtual Serial Ports:
 - ▶ Interactive Shell Interface
 - ▶ Serial Console Channels A and B.

<https://github.com/geertu/teensy3-bcu2>



16 / 39

BCU/2 Shell

Welcome to BFF, the Board Farm Frobnicator!

Valid command are:

Getenv: Get the value of an environment variable
GPIO: Control GPIO
Help: Display this help
History: Show command history
I2c: I2C tools
Key: Control key
Monitor: Monitor power consumption
Power: Control power
Printenv: Print all environment variables
RGB: Show a color
Sample: Sample power
SAveenv: Save all environment variables
SEtenv: Set the value of an environment variable
Test: Test cycle through board features
Version: Display software version

BFF>

SAveenv not yet implemented



17/39

Adding More Boards (Take Two)

Challenges

- ▶ CDC-ACM: 3 USB Endpoints (2 locked) per port
 - ▶ mUSB in TI AM335x SoC: Max. 16 Endpoints
- ⇒ Max. 2 BCU/2 boards per BeagleBone Black

Solution

- ▶ Moxa UPort multiplexes 2/4/8/16 ports over 3 endpoints
- ▶ Added basic support to Teensyduino, based on (behavior of) the Linux driver
- ▶ ~~Alternative: new Linux and Teensy drivers for my own protocol~~

<https://github.com/geertu/teensy3-bcu2/tree/mxu-serial>



18/39

Adding More Boards (Take Three)

Challenges

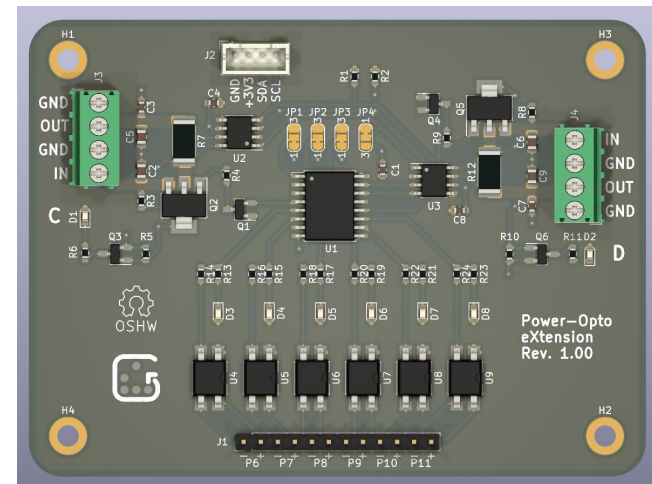
- ▶ BayLibre ACME is no longer available
- ▶ PJRC Teensy 3.2 is no longer available
- ▶ Teensy 4.0 is pin-compatible, but also difficult to get

Solution

- ▶ Drop less-needed parts from BCU/2 (Serial, RGB LEDs)
- ▶ Replace Teensy 3.2 by I2C GPIO Expander
- ▶ Connect to BCU/2 I2C Expansion Connector (or any SBC with I2C, Future Proof)



19/39



<https://github.com/geertu/pcb-pox>

<https://github.com/geertu/teensy3-bcu2/tree/bcu2%2Bpox>



20/39

BCU-POX — Power-Opto eXtension

Challenges

P-Channel MOSFET no longer available after PCB design

Solution

- ▶ Change resistor values controlling gate drive
- ▶ Future: SOIC-8 package is more common than SOT-223?



21 / 39

Current State

- ▶ IKEA Omar Shelving
 - ▶ 14 SBCs and Devboards
 - ▶ 450W PC Power Supply
 - ▶ 2× 16-port 1 Gbps Managed Ethernet Switches
 - ▶ Control Hardware (5Vsb) →
- 2× BeagleBone Black
 - 1× BayLibre ACME
 - 3× Opto-Board
 - 2× BCU/2
 - 1× BCU-POX
 - 2× 10-port USB 2.0 hub
 - 1× Quad USB-serial adapter
 - 1× INA219 for 5Vsb (1.4A idle)

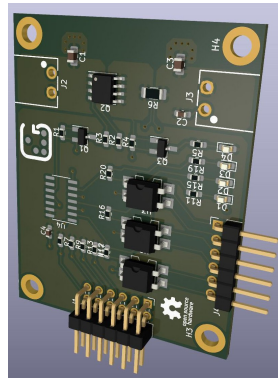


22 / 39

Adding More Boards (Take Four)

Work In Progress

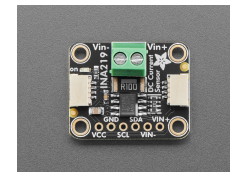
- ▶ Plug-In Boards
 - ▶ Power Control and (optional) Measurement
 - ▶ Opto-Coupler Control
 - ▶ Can be used stand-alone (GPIO)
 - ▶ Bottom row compatible with Pmod™ Interface Type 1 (GPIO)
 - ▶ Multiple variants (e.g. USB power/data switch)
- ▶ Backplane
 - ▶ Accepts multiple Plug-In Boards
 - ▶ GPIO Expander
 - ▶ Actual Power Measurement



23 / 39

Alternatives

- ▶ Remote-Controlled Power Outlets
- ▶ Relay Boards (USB/Ethernet/GPIO)
- ▶ Adafruit INA219 High Side DC Current Sensor Breakout
- ▶ LXA Test Automation Controller



Images © Adafruit, Linux Automation GmbH



24 / 39

My Ideal Board Farm

- ▶ 48VDC power, per-board DC-DC Converter
- ▶ Pluggable Architecture
 - ▶ USB
 - ▶ Ethernet (isolated)
 - ▶ Video In/Out
 - ▶ ...
- ▶ High-Speed Backbone (PCIe?)



25 / 39

Board Farm Software

Collection of Organically-Grown Scripts

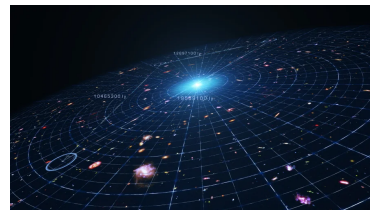
Ugly Web Server written in PERL running on BeagleBone Black



26 / 39

Expanding The Universe

Once you have a board farm,
people start to ask questions ...



- ▶ *Can I set up a board farm myself, too?*
- ▶ *Here's a new board. Can you add that to your farm?*
- ▶ **Can I get access to (a board in) your board farm?**

Image © NASA's Goddard Space Flight Center Conceptual Image Lab



27 / 39

Let's Play a Small Game!

There won't be a Closing Game ☹



Rock



Paper



Scissors

Sorry to disappoint you, but the game is only available at the live conference

Image © Vecteezy



28 / 39

Board FRAM - Board Farm Remote Access Management

Design Decisions / Features

- ▶ Each board has its own UNIX user (cfr. Android apps)
- ▶ Authentication provided by SSH (cfr. Gitolite)[†]
 - ▶ No special tools needed on the client, just **ssh** and **rsync**
`ssh <board>@<remote> <command> ...`
- ▶ Not tied to any specific board-farm management software
 - ▶ Separation between remote access and board farm infrastructure
 - ▶ User provides glue
- ▶ KISS, easy to review for correctness and security
 - ▶ String comparison
 - ▶ Call provided shell functions, scripts, external tools
 - ▶ Bash, 431 LoC (**shellcheck!**)

▶ <https://github.com/geertu/board-fram>

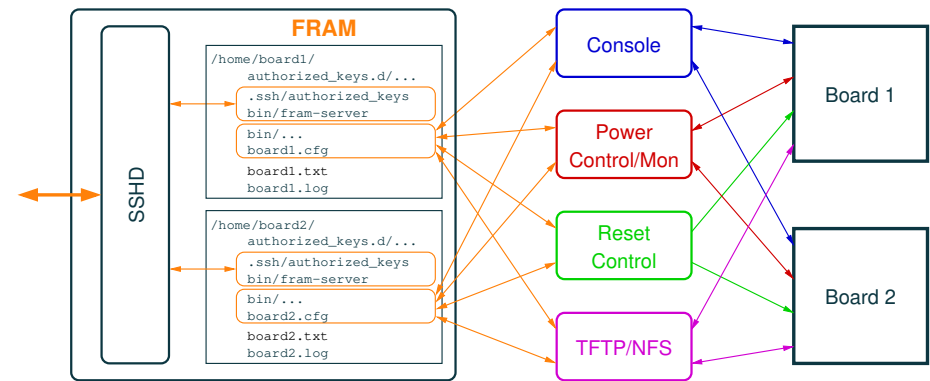
[†]Inspired by <https://github.com/kbingham/lab>



29 / 39

FRAM Architecture

All Computer Science Problems can be solved by adding Yet Another Layer



30 / 39

FRAM User Commands

```
$ ssh h3-salvator-xs@farm.glider.be help
Board FRAM - The Board Farm Remote Access Manager for h3-salvator-xs
```

Valid commands are:

| | |
|-------------------------------|--|
| help | This usage information |
| acc [on off status*] | Control board accessory switch |
| console | Access the board console (use "ssh -t") |
| lock | Lock the board for exclusive access |
| ls | List the boot directory contents |
| power [on off sample status*] | Control board power |
| release | Release the board lock |
| reset | Reset board |
| rsync | Upload files to boot directory (use rsync) |
| ssh-proxy | Open a netcat proxy path to the target |
| status | Show board status |
| wake | Wake board by key |
| wol | Wake board through Wake-on-LAN |



31 / 39

Board Admin Commands

Valid admin commands are:

| | |
|-------------------|--|
| logs | View and monitor the logs |
| offline [msg ...] | Make the board unavailable |
| online | Make the board available again |
| setup <user> | Set up environment for specified user |
| shell | Launch a shell (use "ssh -t") |
| steal | Release then immediately lock (without setup!) |
| unlock | Release the board from any lock |



32 / 39

[...]

FRAM Board Operations

- ▶ To be implemented by **You**
- ▶ Shell functions in `$BOARD.cfg`, scripts, external commands

| | | |
|------------------|----------------------------|--|
| Mandatory | Serial Console Access | <code>console-\$BOARD</code> |
| | External Power Control | <code>power-\$BOARD-on</code> <code>power-\$BOARD-off</code> <code>power-\$BOARD-status</code> |
| | Reset Control | <code>reset-\$BOARD</code> |
| Optional | External Power Measurement | <code>power-\$BOARD-sample</code> |
| | Accessory Switch Control | <code>acc-\$BOARD-on</code> <code>acc-\$BOARD-off</code> <code>acc-\$BOARD-status</code> |
| | Wake-Up Control | <code>wake-\$BOARD</code> <code>wol-\$BOARD</code> |
| | User Environment Set-Up | <code>setup-\$BOARD \$USER</code> |
| | | |



33/39

FRAM Example Config (partial)

```
ADMIN_USERS="geert"

# Operations implemented as shell functions:
function power-h3-salvator-xs-on()
{
    # "power-main" is a script to check the main power status of the farm
    if grep -q 0 <(power-main status | tail -1); then
        echo "Main power is off. Please contact admin."
        exit 1
    fi
    # "power-acme" is a script to control power on BayLibre ACME
    power-acme h3-salvator-xs on
}

# Other operations not implemented here:
# - "console-h3-salvator-xs" is a script to launch screen on the right
#   host with the right parameters
```



34/39

Tips & Tricks

- ▶ `screen -c screenrc -xR -S <board> /dev/serial/by-id/<id> <speed>`
- ▶ Restricted screen session! `examples/screenrc`
- ▶ Base nfsroot + per-user overlays `examples/overlay-setup`
- ▶ Run command as root:
 - ▶ Local: `sudo /etc/sudoers.d/board-farm:`
`<board> ALL = NOPASSWD: /path/to/overlay-setup`
 - ▶ Remote: per-board SSH private key without passphrase
- ▶ Client convenience wrapper `~/.framrc:`
`nickname=board@remote.org`
`fram <nickname> <command> ...`
- ▶ SSH Proxy `~/.ssh/config:`
`match host <board> !exec "host <board>.<domain>"`
`ProxyCommand fram <board> ssh-proxy`



35/39

FRAM Future?

- ▶ Better documentation
- ▶ Console stuff
- ▶ Multi-Media enhancements
- ▶ ...



36/39

- ▶ **Kieran Bingham**, for design brainstorming, development, deployment, feedback,
- ▶ **Niklas Söderlund**, for remote use, feedback, integration into his personal testing framework,
- ▶ **Renesas Electronics Corporation**, for contracting me for upstream Linux kernel work, and for populating the lion's share of my board farm,
- ▶ The **Linux Foundation**, for organizing this conference and giving me the opportunity to present here,
- ▶ The **Linux Kernel Community**, for having so much fun working together towards a common goal.



37 / 39



38 / 39



39 / 39